Journal of Computers Vol. 31 No. 3, 2020, pp. 100-114
doi:10.3966/199115992020063103008

Bluetooth Indoor Positioning and Correction Method Based on Matrix Completion and Compressed Sensing

Gang Yang, Yawen Chai*, Lu Han

School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
845849567@qq.com

Received 31 October 2018; Revised 17 May 2019; Accepted 31 May 2019

Abstract. Based on the received signal strength indication of Bluetooth low energy, a new fingerprint positioning method is proposed, which does not need to collect too many data points to establish fingerprint database and can correct the positioning error in time. In the offline stage, the low-rank matrix completion theory is used to recover a fingerprint database with only a small number of collection points; in the online stage, after clustering and matching the fingerprint database, the localization problem is transformed into compressed sensing model, and the localization result is obtained by solving the optimization problem. Experiment shows that the proposed positioning method can effectively reduce about 40% workload of early data acquisition and reduce the positioning error to 0.7m. Compared with the traditional KNN matching algorithm, the complexity of the algorithm is reduced and the accuracy is improved.

Keywords: Bluetooth low energy, compressed sensing, fingerprint localization, indoor positioning, matrix completion

1 Introduction

For indoor positioning, due to the complex indoor environment and more occlusion, the fingerprint localization method can make good use of these features for positioning. Fingerprint localization method includes two stages: offline data acquisition and online data matching. The density of the collected data during the offline stage determines the positioning accuracy. The more intensive the data acquisition is, the more accurate the positioning results are. However, the acquisition of offline fingerprint data is proportional to the cost of human and material resources. Assuming a 100m *100m area, a data point is collected every 1m. Considering the influence of direction, equipment and pedestrians, the amount of data needed to be collected is undoubtedly huge, and if the indoor environment changes in the upper positioning stage, the fingerprint database needs to be updated in real time to ensure the accuracy of the positioning results. Therefore, how to reduce the workload while maintaining the positioning accuracy and updating the fingerprint database in time is the target of this paper.

Some scholars have done a lot of work to reduce the establishment stage of fingerprint database. In 2013, Laoudias et al. put forward the idea of "crowdsourced" and "swarm intelligence perception" to allow users who hold different devices to establish and feedback fingerprint databases [1-3]. Although the workload of the initial stage of fingerprint database establishment is greatly reduced, this method neglects the user's coordination and the audit mechanism user uploaded data is not complete enough. In 2013, Wu et al. proposed that the user's real location can be obtained by mapping the virtual room to the real geographical location using Hungarian matching algorithm without establishing fingerprint database at the beginning [4], but in the process of mapping, the connectivity and coverage of the real geographic location need to be considered. In 2014, Liu and Wei proposed a method that collecting only part of the fingerprint data and restoring the rest of the data using compressed sensing theory [5-6], but there is no solution to the updating and perfecting of the fingerprint database. In 2017, Zhang et al. proposed using

* Corresponding Author

Newton interpolation and Kriging interpolation to improve the fingerprint database [7-8], but the interpolation method is usually used in the construction and mining fields, the accuracy of restoring fingerprint database is slightly poor, the early variogram is usually selected according to experience, the effect is not ideal. In 2017, Li et al. proposed to improve the fingerprint database by using matrix completion method [9-10], the effect is remarkable, however, the traditional class matching method is still used in the positioning stage, and the algorithm is complicated.

Based on matrix completion theory, a method to improve fingerprint database is proposed. In the early stage, some data points are used to recover the remaining data points, which saves workload. In the later stage, the recovered fingerprint database are clustered and partitioned, and then the positioning results are reduced to two categories by using K nearest neighbor (KNN). The localization problem is transformed into a compressed sensing (CS) model, and the optimization problem is solved to obtain the localization results, which reduces the matching workload. The real-time received signal strength indication (RSSI) obtained by users are stored in the database for real-time updating and correction of fingerprint database. The inverse distance weighted (IDW) interpolation is used to improve the fine fingerprint database and then improve the positioning accuracy. Compared with the traditional clustering matching method KNN, CS-KNN can reduce the workload and improve the positioning accuracy, and improve the problem that fingerprint database can not be updated in time when the environment changes.

2 Theoretical Basis and System Model

2.1 Matrix Completion and CS Theory

Matrix completion comes from the classical Netflix problem, which means filling the missing data in the matrix with incomplete elements, mainly using the low rank of the original data matrix to reconstruct the matrix. Matrix completion is actually the affine rank minimum problem (ARMP) [11]. In special cases under stochastic constraints, ARMP is as follows (1):

$$\begin{gathered} min\ rank\ (X) \\ subject\ to\ X_{ij} = M_{ij}, (i, j) \in \Omega \end{gathered} \tag{1}$$

Where ***X*** represents the matrix to be filled, ***rank(X)*** represents the rank of the matrix to be filled, ***i*** 、***j*** represent row element and column element subscript of matrix. When there is a random mapping relationship, the matrix ***M*** can be restored, it means ***M*** represents the restored matrix.

CS is a method proposed by Donoho et al. in 2004, which can be widely used in compressible signal acquisition [12]. Assuming that the signal ***x*** to be sampled is non-zero at ***k*** times (***k*** is sparse degree), in order to collect the information in ***x***, a set of given waveforms Φ is used to sense ***x***, and a set of measures ***y*** which are far less than the original length of the signal is obtained, then the CS method can recover all the information of the original signal ***x*** by the measure ***y*** which is much smaller than the data of the sampled signal. The process is as follows (2), where $\| x \|_0$ represents the 0 norm of ***x***:

$$\begin{gathered} min \| x \|_0 \\ s.t.\ y = \Phi x \end{gathered} \tag{2}$$

The 0 norm problem is usually approximated by the 1 norm problem as follows (3), where $\| x \|_1$ represents the 1 norm of ***x***:

$$\begin{gathered} min \| x \|_1 \\ s.t.\ y = \Phi x \end{gathered} \tag{3}$$

In practical applications, the signal ***x*** is not usually sparse, but the transform coefficients θ on a basis ψ are sparse or compressible, so ***x*** can be expressed as follows (4):

$$x = \psi \theta \tag{4}$$

Determined transformation coefficients the θ are sparse, or very few coefficients contain almost all the energy, so the CS model can be written as (5):

$$x = \Phi \psi \theta \tag{5}$$

Matrix completion problem is alike CS theory, many viewpoints and ideas have inheritance and consistency, and they are closely related. Matrix completion mainly utilizes the low rank or approximate low rank characteristics of the original matrix to recover the matrix with missing elements, and CS reconstrucs the original signal mainly utilizes the sparse characteristics of the signal. At the same time, E Candes et al. proposed the restricted isometry property (RIP) for CS, pointing out that the higher degree of incoherence between perceptual matrix Φ and transformation matrix ψ , the greater the possibility of restoring the original signal [13]. Similarly to the sparsity of CS, if a matrix is a low-rank matrix or a matrix with approximate low-rank, the problem of matrix completion can be solved by solving the above-mentioned optimization problem, because the rank of the matrix remains unchanged before and after completion, and whether the matrix is low rank requires singular value decomposition (SVD) of the matrix to validate.

2.2 SVD and Singular Value Threshold

Compared with other eigenvalue decompositions method, SVD is robust from a mathematical point of view and can decompose any matrix uniquely. The SVD for matrix *A* is as follows (6):

$$A = \sum_{i=1}^{n} u_i \sigma_i v_i^T \tag{6}$$

Where *u* represents the left singular vector, the same as the number of rows of *A*, *v* represents the right singular vector, σ is a diagonal matrix, and the elements on the diagonal are called the singular values of *A*. In the order from large to small, the number of non-zero singular values is the same as the rank of *A*. In 1985, Laub refered to that a matrix after SVD, if there is $\sigma_1^2 / \sigma_2^2 \gg 1$, then it can be considered that the useful information of the matrix is concentrated on the vector $u_1 \sigma_1 v_1$ [14]; Similarly, in 2005, Zhu refered to that the useful signals in the matrix are mainly reflected by the larger singular values of the first *r* [15]; Zang indicated that if the data to be processed have good time stationarity and the cumulative contribution rate of the first *L* maximum singular values is more than 90%, the data can be proved to have good low rank [16]. Qian mentioned that in engineering applications, the optimal dimension is basically generated in a neighborhood of *N*/2 (where ***N*** stands for signal length) by analyzing a number of source signals of different lengths and frequencies [17], that is to say, the singular value of *N*/2 is selected to restore the original matrix with probability. In this paper, after SVD of the RSSI used in the positioning, the conclusion can be drawn that the fingerprint database can be improved by matrix completion theory. Four Bluetooth base stations broadcast to the outside world. The receiver receives the RSSI matrix of 7*7 according to the fingerprint database construction mode. Four 7*7 matrices are decomposed into SVD respectively. The results are as Fig. 1.

As can be seen from Fig. 1, the maximum singular value of each base station is far greater than the second singular value and other singular values. Among the 7 singular values, the cumulative contribution rate of the first 3 singular values is as high as 95%, which can reflect most of the information of the matrix, so it is suitable for matrix completion theory.

When a matrix has low rank or approximate low rank properties, similar to compressed sensing, it is possible to replace the rank with the kernel norm, i.e. the l_1 norm instead of the l_0 norm. Then the problem is changed to the following (7):

$$\begin{gathered} min \| x \|_1 \\ subject\ to\ M_{ij}, (i, j) \in \Omega \end{gathered} \tag{7}$$

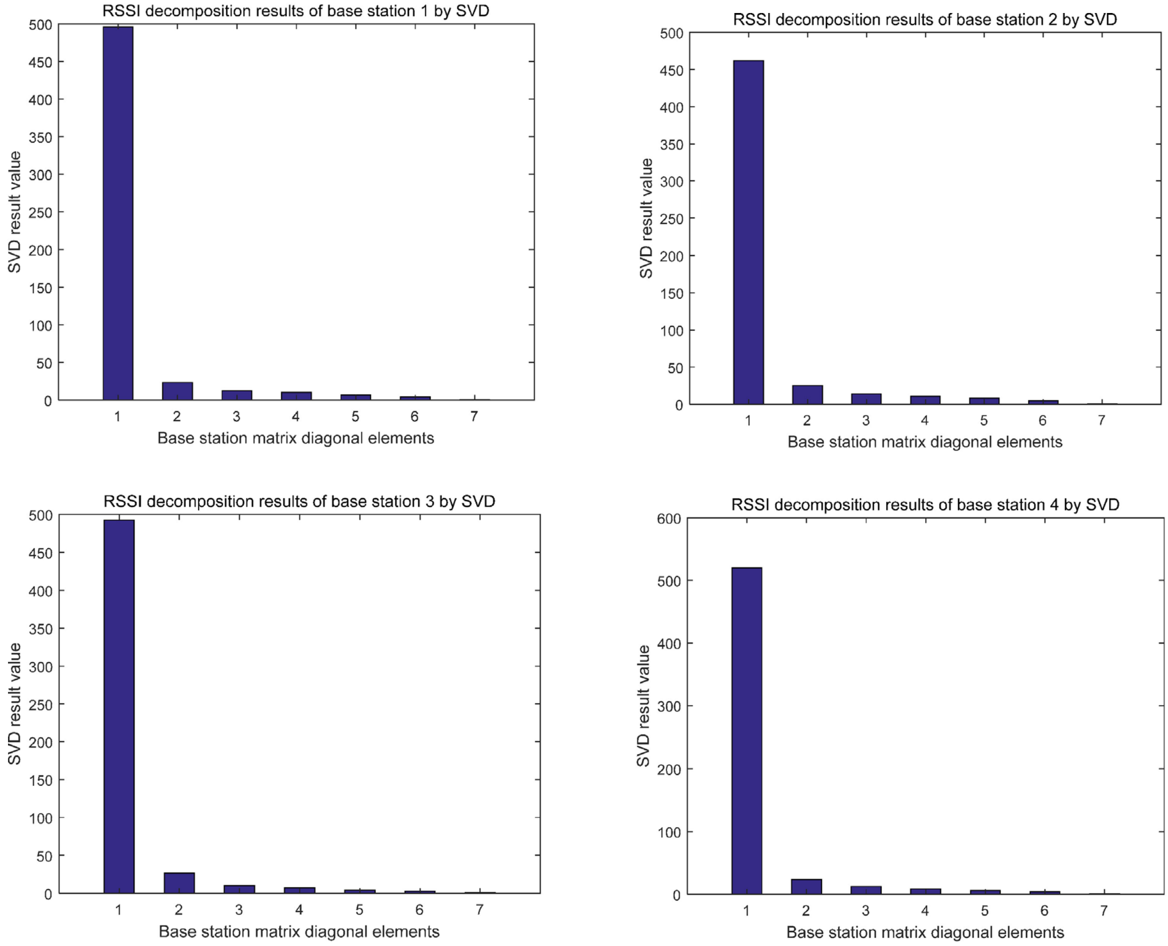

Fig. 1. SVD decomposition results of RSSI matrix received by each base station

Chen proved that problem (1) and problem (7) have the same solution under non-convex conditions [18]. CANDES proposed that the sampling method of matrix completion requires that the sampling model of matrix is Bernoulli model, and that the elements obtained should be distributed in every row and column of matrix [19]. For this kind of convex optimization problem, some algorithms are given, such as Singular Value Thresholding (SVT) algorithm. This algorithm is a simple matrix completion method proposed by Cai et al. [20]. It is an iterative algorithm. The key of iteration is SVD of matrix.

2.3 System Model and Algorithm Flow

The process of fingerprint localization based on Bluetooth low energy is as follows: firstly, the fingerprint database with only a small amount of RSSI is improved by matrix completion theory. Then the improved fingerprint database is clustered by K-means and classified by KNN. Later the localization problem is transformed into CS model. The reconstruction algorithm basis pursuit (BP) solves the final positioning, then the user's real-time location data and location information are stored in the database to improve the accuracy of subsequent positioning. The specific process is as Fig. 2.

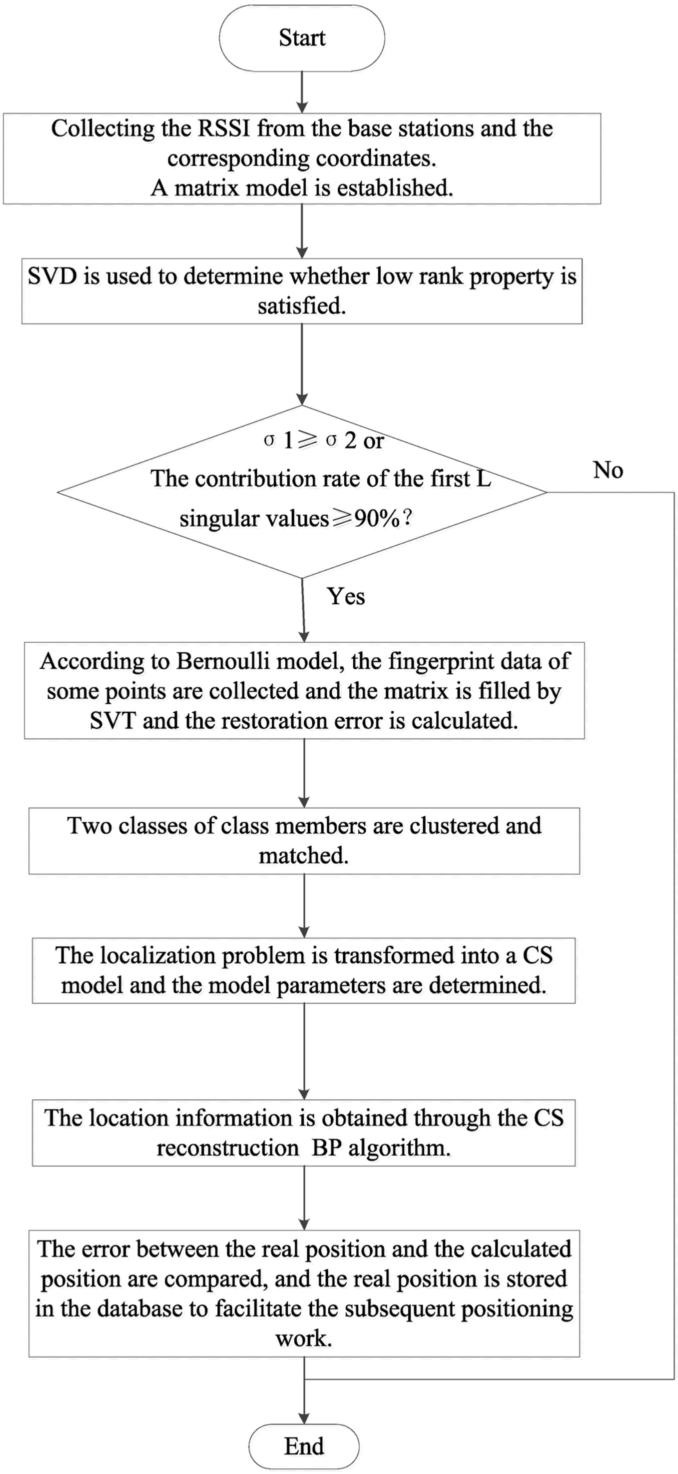

Fig. 2. Flow chart of the whole process

3 Localization Process Based on Matrix Completion and CS

3.1 Matrix Completion and Clustering Matching

The fingerprint database matrix of base station 1 is selected, the original matrix is defined as ***M***, the incomplete matrix to be filled is ***A***, the filled matrix is ***X*** and the error matrix is ***W***. The sampling rate of matrix ***M*** is 60% randomly to recover the remaining 40% data. The results of matrix completion are as Table 1 to Table 4 (Unit /dBm).

Table 1. Fingerprint matrix ***A*** with some unknown values

-77.9245	0	-77.5281	-79.1183	0	-55.1620	-73.2138
-76.0781	0	0	-72.3117	0	0	-63.5711
0	-74.8710	0	-67.1399	-62.1038	-62.5402	-73.8138
0	0	-64.6166	-70.0357	0	-67.7472	-68.0184
-78.3079	-68.0176	0	-69.2883	0	-68.3500	0
0	-65.8516	0	-64.9645	-66.5018	-74.2932	0
-79.6291	0	-67.6737	-71.4692	0	-69.9741	0

Table 2. Filled fingerprint matrix ***X***

-77.9245	**-80.1229**	-77.5281	-79.1183	**-61.1265**	-55.1620	-73.2138
-76.0781	**-69.8128**	**-72.9448**	-72.3117	**-55.6279**	**-62.4563**	-63.5711
-80.7129	-74.8710	**-68.2519**	-67.1399	-62.1038	-62.5402	-73.8138
-75.1289	**-78.2675**	-64.6166	-70.0357	**-59.6775**	-67.7472	-68.0184
-78.3079	-68.0176	**-70.4599**	-69.2883	**-65.1129**	-68.3500	**-74.6377**
-79.8532	-65.8516	**-70.9241**	-64.9645	-66.5018	-74.2932	**-80.1965**
-79.6291	**-78.2698**	-67.6737	-71.4692	**-65.9536**	-69.9741	**-70.6921**

Note. The filling results are shown in bold form.

Table 3. The original complete fingerprint matrix ***M***

-77.9245	-82.38120	-77.5281	-79.1183	-62.4125	-55.1620	-73.2138
-76.0781	-70.9060	-74.4013	-72.3117	-54.5948	-63.3843	-63.5711
-81.6470	-74.8710	-68.9436	-67.1399	-62.1038	-62.5402	-73.8138
-76.0327	-79.2119	-64.6166	-70.0357	-60.1778	-67.7472	-68.0184
-78.3079	-68.0176	-71.7139	-69.2883	-67.8763	-68.3500	-70.5257
-80.4506	-65.8516	-69.5888	-64.9645	-66.5018	-74.2932	-79.6715
-79.6291	-77.6989	-67.6737	-71.4692	-66.6162	-69.9741	-72.8937

Table 4. Error matrix ***W***

0	0.0274	0	0	0.0206	0	0
0	0.0154	0.0196	0	0.0189	0.0146	0
0.0114	0	0.0100	0	0	0	0
0.0119	0.0199	0	0	0.0083	0	0
0	0	0.0175	0	0.0407	0	0.0583
0.0074	0	0.0192	0	0	0	0.0066
0	0.0073	0	0	0.0099	0	0.0302

The average error of four base station filled matrices calculated by MATLAB is 1.84%. It shows that the matrix completion theory can reconstruct the original fingerprint database more accurately. Then K-means clustering is performed for the matrix restored by SVT algorithm. K-means clustering is a typical prototype-based object function clustering method. According to the distance from the data point to the prototype (usually Euclidean distance) as the optimization objective function, the adjustment rule of iterative operation is obtained by using the method of function extremum [21]. In this paper, the RSSI in fingerprint database are clustered into five classes according to K-means algorithm rules. Because it is easy to fall into the local optimal solution in the clustering process, this paper uses the experiment of changing the initial clustering center many times to get the best clustering result.

After clustering, KNN algorithm is used to match the two classes according to the real-time RSSI obtained by the users. The purpose is to prevent the huge positioning error caused by the jump of the RSSI, reduce the positioning error and concentrate the error to the maximum in two categories. The core idea of KNN algorithm is to select *K* spaces which are closest to the Euclidean distance of the sample data and take the average value as the final matching result. In this localization process, the real-time user data is compared with the clustering centers after K-means clustering, and two clustering centers with the smallest Euclidean distance are selected as matching results.

3.2 CS Localization and Fingerprint Database Correction

At the end of the previous section, the localization results are theoretically limited to two classes. In this section, the following localization process is transformed into a CS model to solve. Assume there are Q class members in these two classes in Section 3.1. Considering the localization result from space, the location result θ is unique and sparse in space, and the result is θ = [0, 0, …, 1, 0, …0], $\theta(Q)=1$, indicating that θ can only be in one location at a certain time, and 0 in other locations. Therefore, the matrix ψ is defined as a matrix of L*N, where L represents the number of base stations and Q is all the class members of the two classes in the clustering result of 3.1 section. Because $y=\Phi\psi\theta$ requires that Φ is uncorrelated with ψ, Φ is defined as a L*L random observation matrix, y is a set of observations, which is represented by the RSSI obtained from L base stations in the positioning phase. Through the BP algorithm reconstruction, the final location result ψ is obtained. BP reconstruction algorithm transforms the non-convex problem into a convex problem to find the approximation of the signal, and uses l_1 norm instead of l_0 norm to solve the optimization problem, so as to use linear programming method to solve [22].

After getting the positioning results, this paper hopes to use the positioning results for the future positioning work. However, since the user has only one set of real-time positioning data, which has not been subjected to multiple acquisitions, Gaussian filtering, and average operation, the fluctuation and randomness of the positioning data is large, which makes the authenticity of the positioning data is not enough to be used for positioning. So the paper takes multiple positioning after taking the root mean square (RMS) value for indoor environment changes when fingerprint database updates. At the same time, considering that the fingerprint database is established using grid method to collect data, using inverse distance weighted (IDW) interpolation to expand the fingerprint database grid, narrow the sampling interval, in the subsequent positioning process it can reduce the positioning range and improve the positioning accuracy. IDW takes the distance between the interpolation point and the sample point as the weighted average, and the closer to the interpolation point, the greater the weight given by the sample point. Considering the inverse relationship between RSSI and distance, IDW algorithm is used to interpolate the fingerprint database. In summary, the specific process after obtaining the positioning result is as follows:

4 Experimental Simulation and Result Analysis

4.1 Experiment Environment and Equipment Introduction

The experiment environment selected in this paper is a laboratory in the No. 3 Experimental Building of Xi'an University of Posts and Telecommunications. The laboratory is a square area of about 10 meters in length and width. There are desks, computers, platform, projectors, air conditioning and corridors in the laboratory. The indoor environment is more shaded and more complex. The signal transmitter uses Bluetooth ibeacon of Ghostyu Internet of Things Company as the base station, the model is ibc41, and the built-in chip is TI CC2541. The transmit power is set as 4dBm, the broadcast interval is set as 800ms. The receiver uses Launch IOT development board of Ghostyu Internet of Things Company, built-in Bluetooth processor is CC2640 of TI Company, and embedded development is carried out in C language. The laboratory is divided into 7 x 7 small rectangular areas and the RSSI is collected. According to the layout principle of base stations in indoor positioning proposed in [23], by judging the relationship between the relationship of indoor environment, signal transmission power and RSSI, the base stations are arranged as follows: 1, 2, 3 and 4 represent four different base stations, and each base station corresponds to the MAC address one by one as Fig. 4.

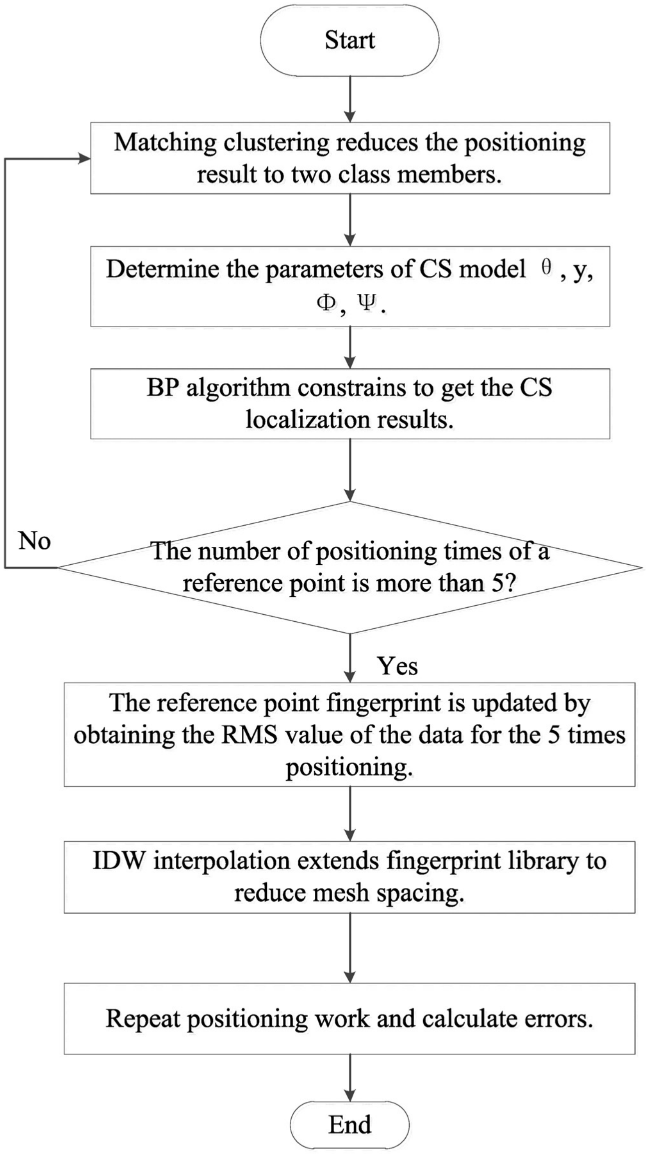

Fig. 3. Corrective process flow chart

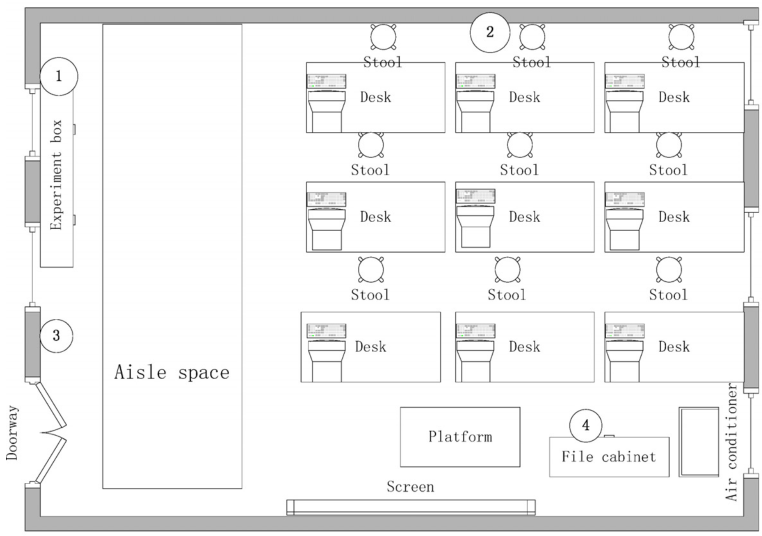

Fig. 4. Indoor positioning experiment environment

For the receiver of the development board, part of the programming language is set up in C language as follows:

```
int i=0, i_2=0, i_3=0, i_4=0;
// Defining 4 base station variables
char mac1 [50], mac2 [50], mac3 [50], mac4 [50];
// Define the MAC address of 4 base stations
int RSSI1, RSSI2, RSSI3, RSSI4;
// Defining signal strength of 4 base stations
                  if((strcmp(Util_convertBdAddr2Str(pEvent-
  >deviceInfo.addr),"0x3CA308AC870F")==0))
{
memcpy(mac2, Util_convertBdAddr2Str(pEvent->deviceInfo.addr),
strlen(Util_convertBdAddr2Str(pEvent->deviceInfo.addr)));
RSSI2=pEvent->deviceInfo.rssi;
}
// The MAC address of each base station corresponds to the base
station number to get the RSSI value.
TaskUARTdoWrite(NULL, NULL, "MAC=%s,%d", mac1, RSSI1);
TaskUARTdoWrite(NULL, NULL, "MAC=%s,%d", mac2, RSSI2);
TaskUARTdoWrite(NULL, NULL, "MAC=%s,%d", mac3, RSSI3);
 TaskUARTdoWrite(NULL, NULL, "MAC=%s,%d\r\n", mac4, RSSI4);
// Output the signal strength value corresponding to the base station
according to the MAC address.
```

The RSSI in the same place fluctuates slightly with time as shown in Fig. 5, the RSSI collected by four base stations is represented by different symbols. The RSSI of one base station is shown in Table 5. For this purpose, the signals collected from each point for many times are filtered by Gauss filter in MATLAB, and then the average value is obtained. The result is shown in Fig. 6.

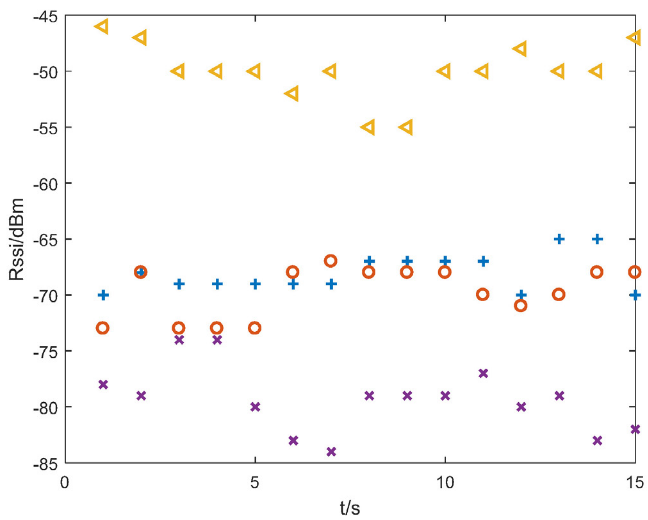

Fig. 5. RSSI results of 4 base stations at certain points over time

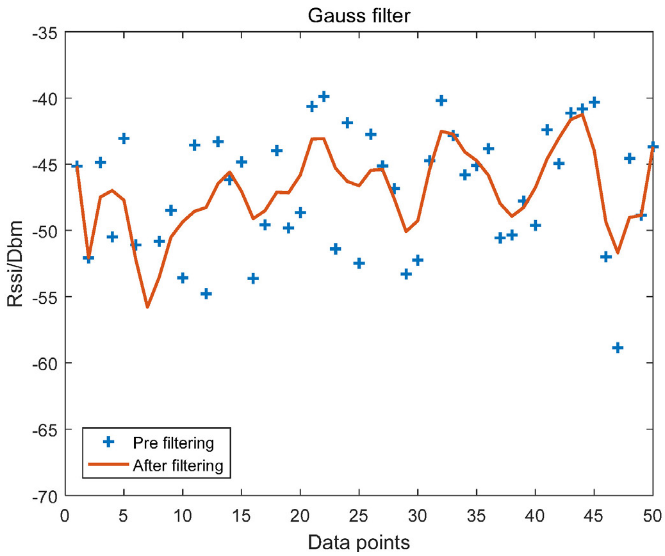

Fig. 6. RSSI Gauss filtering before and after of a base station

Table 5. The MAC address of one of the base stations and the raw data of some measured RSSI (in dBm)

MAC=0x3CA308AC7365	-71
MAC=0x3CA308AC7365	-71
MAC=0x3CA308AC7365	-71
MAC=0x3CA308AC7365	-72
MAC=0x3CA308AC7365	-72
MAC=0x3CA308AC7365	-72
MAC=0x3CA308AC7365	-76
MAC=0x3CA308AC7365	-73
MAC=0x3CA308AC7365	-73
MAC=0x3CA308AC7365	-74
MAC=0x3CA308AC7365	-77
MAC=0x3CA308AC7365	-77

In Fig. 6, the "+" represents the data before Gaussian filtering, and the line represents the data after filtering. The fluctuation phase is slower than that before filtering, which proves the validity of Gaussian filtering. The RSSI of the acquisition point and the position coordinate information measured by the laser rangefinder are stored in the database for subsequent matching and positioning.

4.2 Cluster Matching Positioning and Correction Simulation

After the fingerprint database filled, the K-means clustering algorithm parameters are set and the clustering results are as follows:

In Fig. 7, different symbol represents different classes. In this figure, circles, oblique triangles, regular triangles, plus signs and asterisks represent five kinds of members, among circles class1, 9 class members; oblique triangles class2, 16 class members; regular triangles class3, 9 class members; plus signs class4, 10 class members; asterisks class 5, there are 5 members of the class, a total of 49 members of the class are collected at 7 * 7 points.

After the clustering of the RSSI matrix, the contrast algorithm is used to simulate the RSSI matrix. One is the traditional KNN matching algorithm, which sets K to 5. Five sets of coordinates are selected from the fingerprint database matrix which are closest to the current user RSSI, and then the final position is obtained by calculating the average coordinate value. The other is the algorithm based on CS proposed in this paper which called CS-KNN. Firstly, KNN is used to shrink the location results into two classes, and then BP algorithm based on CS is used to solve the optimization problem to get the final localization results.

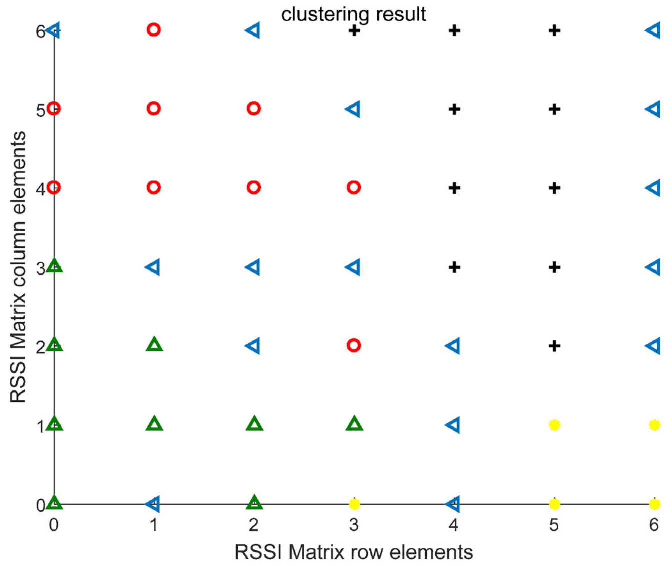

Fig. 7. K-means clustering results

Traditional KNN algorithm. The RSSI at the test point (1, 1) is selected and compared with 49 elements in the fingerprint database. The final five coordinate values are selected as shown in Fig. 8.

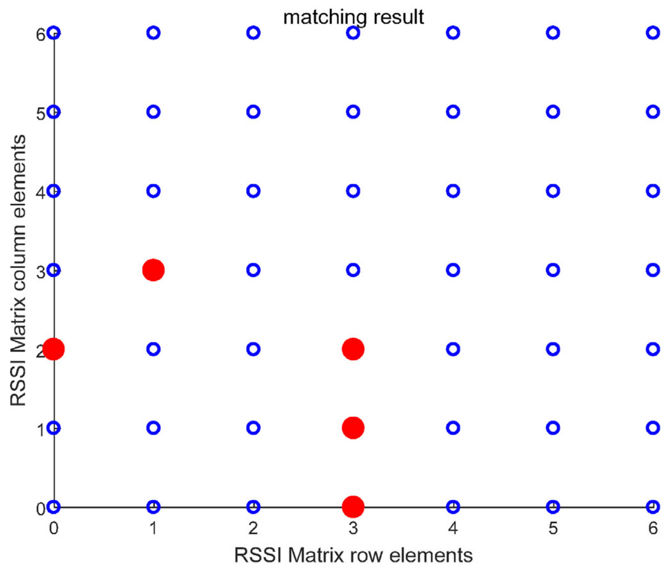

Fig. 8. KNN matching results

As can be seen in Fig. 8, the five points matched by KNN are (0, 2), (1, 3), (3, 0), (3, 1), (3, 2). The final positioning results obtained by averaging the five points are (2, 1.6). Compared with the real coordinates (1, 1), the error is 1.1662m.

CS-KNN Algorithms. The RSSI at the test point (1, 1) is selected to match the KNN algorithm. Set the parameters of KNN algorithm: the number of base stations used for positioning is 4, K is set as 2, the matching results of the two classes are class2 and class3 in MATLAB, a total of 25 class members. The RSSI at the test data point (1, 1) is shown in Fig. 7, which belongs to the class 3. The matching results show that the matching algorithm is correct.

Then the CS model solves the localization result, choosing y as the signal observation value at (1, 1), ψ as the class member of class 2 and class3, and uses BP algorithm to restrict the uncorrelation of ψ and Φ. The simulation test results show that the position coordinates of the class member where x is located (1, 2), and the localization result reflects the real position coordinates (1, 1) in Fig. 9.

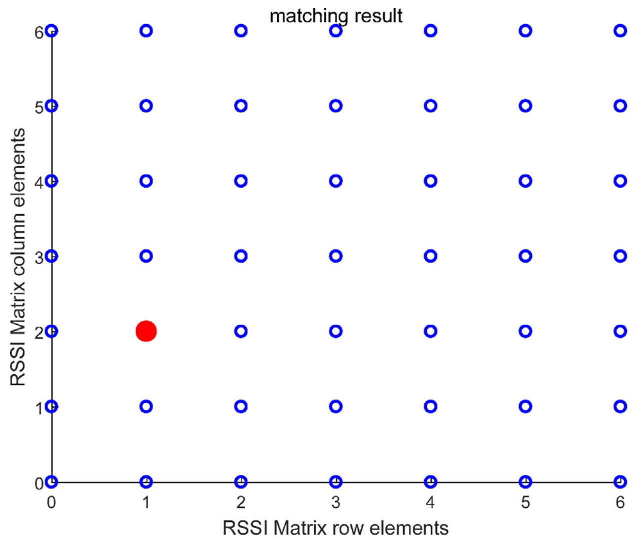

Fig. 9. CS-KNN matching results

In Fig. 9, the result of compressive sensing localization is at (1, 2) points. Compared with the coordinates of real location (1, 1), the error of localization result reflected in real location is 1m.

The results of KNN and CS-KNN are put in a graph, and the comparison with the real location is shown in Fig. 10.

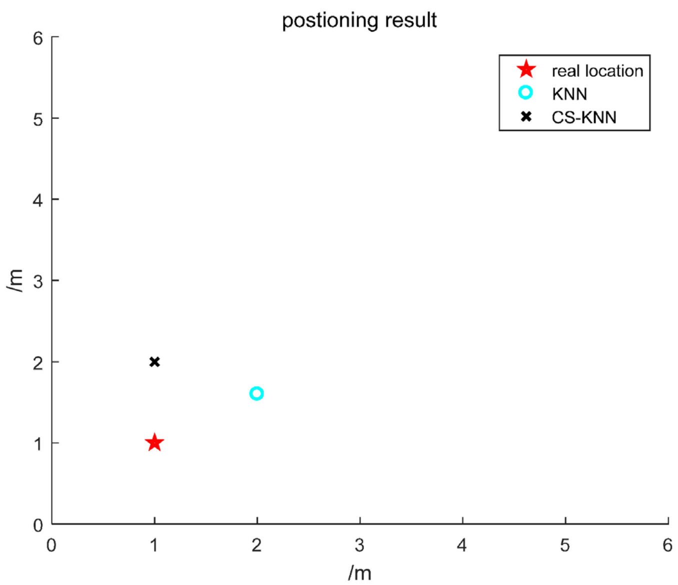

Fig. 10. positioning results of two algorithms

Including (1, 1), the paper has tested (1, 1), (1, 2), (2, 2), (2, 3), (2, 6), (3, 3), (3, 5), (4, 2), (4, 4), (4, 6), (5, 1), (5, 3), (5, 5), (6, 2), (6, 4), (6, 6) sixteen points. The error simulation results with the real geographical location are as Fig. 11.

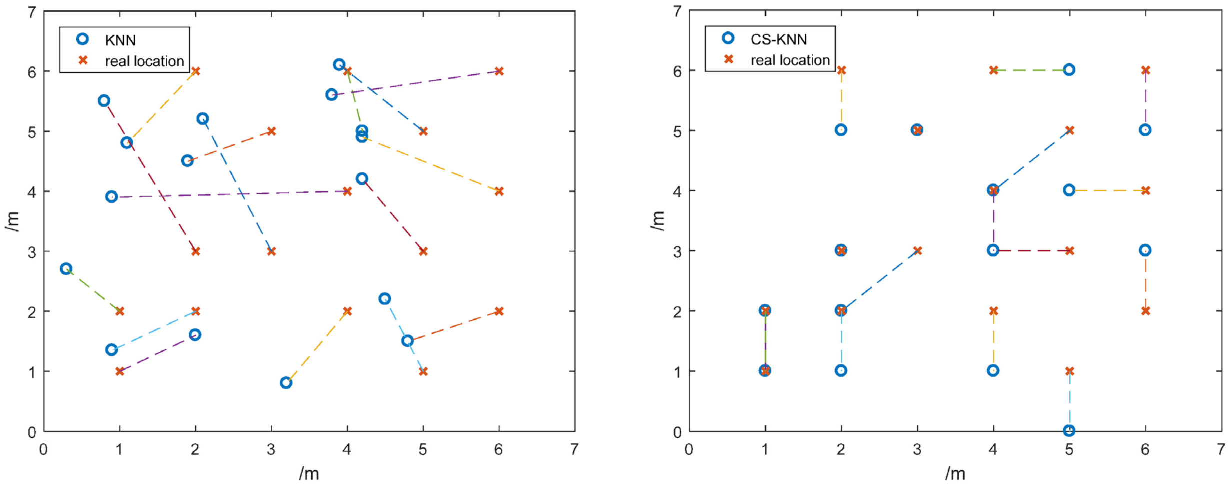

Fig. 11. Comparison of position results and real location of two algorithms

The symbol Λ in Fig. 11 represent the real coordinates of the 16 points tested above, and the circles represent the computational results of positioning. The coordinates of the test points and the results of positioning are connected with dotted lines for easy viewing. The left side is the comparison between the calculated results of KNN algorithm and the real coordinates, and the right side is the comparison between the calculated results of CS-KNN and the real coordinates. The average positioning error of the 16 points mentioned above is calculated by MATLAB. The KNN algorithm is 1.6689m and the CS-KNN algorithm is 0.9268m. The validity of localization algorithm based on CS is proved.

After the location result is obtained, the experiment selects a random point in the fingerprint database (2, 4) and takes RMS of 5 groups of test data. Before the correction, the fingerprint at this coordinate is [-62.1038, -70.7019, -68.8412, -75.7366], and the revised fingerprint is [-58.129, -70.1725, -65.4722, -75.632]. Compared with the previous fingerprint, there is a certain change in the database, indicating that the indoor environment has changed slightly. At the same time, IDW interpolation is applied to the fingerprint database as Table 6.

Table 6. RSSI data of base station 1 fingerprint database after interpolation

-77.9245	**-73.3681**	-82.3812	**-72.8269**	-77.5281	**-71.7575**	-79.1183	**-69.6117**	-62.4125	**-67.3101**	-55.162	**-67.8232**	-73.2138
-73.1258	**-72.9079**	**-72.7437**	**-72.2735**	**-71.9109**	**-71.249**	**-70.5738**	**-69.1361**	**-67.6773**	**-67.377**	**-67.1347**	**-67.7964**	**-68.407**
-76.0781	**-72.6226**	-70.906	**-71.8263**	-74.4013	**-70.8467**	-72.3117	**-68.6333**	-54.5948	**-67.2195**	-63.3843	**-67.7451**	-63.5711
-73.1933	**-72.6353**	**-72.0822**	**-71.5304**	**-70.9895**	**-70.3553**	**-69.6249**	**-68.6519**	**-67.625**	**-67.626**	**-67.6745**	**-68.168**	**-68.6179**
-81.647	**-72.8511**	-74.871	**-71.3582**	-68.9436	**-69.9789**	-67.1399	**-68.598**	-62.1038	**-67.7946**	-62.5402	**-68.5264**	-73.8138
-73.1118	**-72.6483**	**-72.1858**	**-71.2222**	**-70.319**	**-69.8793**	**-69.3924**	**-68.772**	**-68.11**	**-68.1965**	**-68.3303**	**-68.7802**	**-69.2205**
-76.0327	**-72.6026**	-79.2119	**-71.1268**	-64.6166	**-69.8051**	-70.0357	**-68.8536**	-60.1778	**-68.4189**	-67.7472	**-68.9954**	-68.0184
-72.7316	**-72.207**	**-71.6644**	**-70.9566**	**-70.3083**	**-69.9647**	**-69.6254**	**-69.1865**	**-68.7671**	**-68.9165**	**-69.1109**	**-69.3532**	**-69.5418**
-78.3079	**-71.9835**	-68.0176	**-70.7836**	-71.7139	**-70.0759**	-69.2883	**-69.4012**	-67.8763	**-69.3024**	-68.35	**-69.7672**	-70.5257
-72.8185	**-71.8851**	**-70.9721**	**-70.6823**	**-70.4093**	**-70.0129**	**-69.6057**	**-69.4719**	**-69.4007**	**-69.6645**	**-70.0307**	**-70.356**	**-70.7361**
-80.4506	**-71.9824**	-65.8516	**-70.5926**	-69.5888	**-69.9046**	-64.9645	**-69.4183**	-66.5018	**-69.9185**	-74.2932	**-70.9798**	-79.6715
-73.0363	**-72.228**	**-71.4824**	**-70.8813**	**-70.3653**	**-70.0866**	**-69.8157**	**-69.6831**	**-69.5927**	**-69.9681**	**-70.4434**	**-70.8118**	**-71.2387**
-79.6291	**-72.6272**	-77.6989	**-71.1771**	-67.6737	**-70.252**	-71.4692	**-69.89**	-66.6162	**-69.9314**	-69.9741	**-70.6422**	-72.8937

(*Note*. bold data is the result of interpolation)

Although the interpolation results are not necessarily accurate compared with the real measured data, it can increase the number of matches in the positioning process, and expand the 7*7 test data to 13*13 grid data, reflecting that the average sampling interval of 1m is reduced to 0.5m in the real geographic location, and the positioning results are limited to a smaller space in order to obtain the positioning. After the result, the RMS operation can be used to rectify the fingerprint database, and the positioning accuracy can be further improved, the simulation result is showed in Fig. 12.

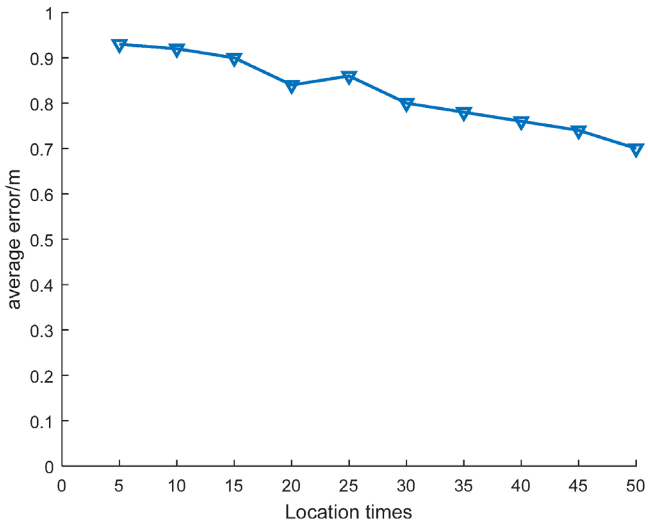

Fig. 12. Relationship between positioning times and error

From the simulation results, it can be seen that when the fingerprint database data used for positioning are frequently corrected and the number of fingerprint databases increases, the positioning error decreases gradually, and in the subsequent positioning process, with the change of environment, the fingerprint database can be constantly updated, while the positioning accuracy improves continuously, and the initial positioning accuracy of 0.9268m is reduced to 0.7m.

5 Conclusion

In this paper, an indoor localization algorithm based on Bluetooth low-energy RSSI is proposed. Firstly, a part of the RSSI data is filled with low-rank matrix completion theory. Then K-means clustering and KNN matching are performed on the filled fingerprint database. Next the localization problem is transformed into CS model for the result. Finally the RMS operation and IDW interpolation are used to correct and improve the fingerprint database. Experiment shows that the workload is reduced by 40% when the error is 1.84% in the offline stage. Compared with traditional KNN algorithm, the CS-KNN algorithm's positioning accuracy is 44.47% higher than traditional KNN algorithm. In a word, the proposed method of this paper saves the workload of traditional fingerprint algorithm, and can correct the data in the fingerprint database in time. With the increase of localization times, the accuracy of the localization results is further improved, the minimum error in experiment is 0.7m.

Acknowledgements

This paper was supported by the Shaanxi Provincial Social Development Science and Technology Project (No. 2016SF-410) and the Shaanxi Provincial International Science and Technology Cooperation Project (No. 2018KW-025).

Reference

[1] C. Laoudias, D. Zeinalipour-Yazti, C.G. Panayiotou, Crowdsourced indoor localization for diverse devices through radiomap fusion, in: Proc. 2013 International Conference on Indoor Positioning & Indoor Navigation, 2013.

[2] Q.Y Chen, Research on Fingerprint Crowdsourcing-based Indoor Localization, Huazhong University of Science and Technology, Wuhan, China, 2016.

[3] X.C. Liu, S. Zhang, H.H. Lu, X.K. Lin, Method for efficiently constructing and updating radiomap of fingerprint positioning, in: Proc. 2011 IEEE Globecom Workshops, 2011.

[4] C.S. Wu, Z. Yang, Y.H. Liu, W. Xi, WILL: wireless indoor localization without site survey, IEEE Transactions on Parallel & Distributed Systems 24(4)(2013) 839-848.

[5] S.L. Liu, X.Z. Tan, Z.H. Jia, Research on WiFi indoor positioning system based on compressed sensing signal reconstruction, Laser Journal 35(09)(2014) 82-85.

[6] D.H. Wei, Research on fingerprint location technology based on compressed sensing, Beijing University of Posts and Telecommunications, Beijing, China, 2013.

[7] D. Zhou, F.M. Hu, H.P. Hu, Z. Zhang, RFID indoor location based on Newton interpolation, Electronic Sci & Tech 26 (9)(2013) 34-35.

[8] X.Q. Zhang, F. Fang, K.Z. Ying, K. Fang, Q.Z. Chen, K.J. Mao, Design and implement of an interpolation indoor fingerprint-based localization system, Chinese Journal of Sensors and Actuators 30 (4)(2017) 596-602.

[9] L.N. Li, W.H. Li, H.X. You, Y. Wang, High efficient construction of location fingerprint database based on matrix completion improved by backtracking search optimization, Journal of Computer Applications 37(7)(2017) 1893-1899.

[10] W. Zhou, An Efficient Radio Map Construction Method for WIFI Positing, Beijing University of Technology, Beijing, China, 2013.

[11] P. Jain, R. Meka, I.S. Dhillon, Rank minimization via online learning, in: Proc. the 25th international conference on Machine learning, 2008.

[12] D.L. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52(4)(2006) 1289-1306.

[13] E.J. Candes, J. Emmanuel, The restricted isometry property and its implications for compressed sensing, Comptes rendus – Mathématique 346(9-10)(2008) 589-592.

[14] A. Laub, Numerical linear algebra aspects of control design computations, IEEE Transactions on Automatic Control 30(2)(2985) 97-108.

[15] X.Z. Zhao, B.Y. Ye, T.J. Chen, Signal decomposition algorithm based on varging matrix structure singular value decomposition, Journal of Vibration, measurement & Diagnosis 38(06)(2018) 1096-1102.

[16] F. Zang, Method of restoring meteorological data using low-rank matrix filling technique, Computer Applications and Software 34(9)(2017) 322-327.

[17] Z. Qian, L. Cheng, Y. Li, Noise reduction method based on singular value decomposition, Journal of Vibration Measurement & Diagnosis 31(4)(2011) 459-463.

[18] F.F. Chen, A Study of the application of SVT algorithm in Netflix problem, Tsinghua University, Beijing, China, 2011.

[19]E. Candes, B. Recht, Exact matrix completion via convex optimization, Communications of the ACM 55(6)(2012) 111-119.

[20] J.F. Cai, E.J. Candes, A singular value thresholding algorithm for matrix completion, Siam Journal on Optimization 20(4)(2008) 1956-1982.

[21] A. He, Y. Guo, Z. Zhang, S. Wang, Research on Gaussian mixture model based on K-mean clustering, Journal of Guangdong University of Petrochemical Technology 28(6)(2018) 73-77.

[22] Y.P. Du, Y.Y. Liu. Simulation analysis of compressive sensing reconstruction algorithm, Electro-optic Technology Application 33(5)(2018) 37-40.

[23] L. Guo, J. Zhen, D.H. Yang, Effects of base station layout scheme on Bluetooth fingerprint position, Journal of Navigation and Positioning 6(2)(2018) 93-96.